National College of Ireland

Performance Evaluation of ML-DSA–Based Post-Quantum Authentication in IoT Healthcare Using Compression Optimisations

MSc Research Project
Msc CyberSecurity

Sylesh Sathish Kumar
Student ID: x23408880

School of Computing
National College of Ireland

Supervisor: Eric Gyamfi

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Sylesh Sathish Kumar
Student ID:	x23408880
Programme:	Msc CyberSecurity
Year:	2025
Module:	MSc Research Project
Supervisor:	Eric Gyamfi
Submission Due Date:	11/12/2025
Project Title:	Performance Evaluation of ML-DSA–Based Post-Quantum Authentication in IoT Healthcare Using Compression Optimisations
Word Count:	6020
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sylesh Sathish Kumar
Date:	10th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Performance Evaluation of ML-DSA–Based Post-Quantum Authentication in IoT Healthcare Using Compression Optimisations

Sylesh Sathish Kumar
x23408880

Abstract

The rapid development of quantum computing poses a risk of compromising the classic algorithms in cryptography including the RSA and the ECC, and exposing connected systems, including the Internet of Things (IoT) to mass data theft and harvest-now, decrypt-later attacks. To address this issue, the project deploys and validates the ML-DSA-44, a NIST-recommended post-quantum digital signature algorithm, as part of a simulated IoT environment, consisting of sensor, gateway, and verification layers. The paper also includes data and certificate compression based on Zlib algorithm to minimize the transmission overhead and energy consumption. Experimental comparisons between payloads ranging between 200-5000 KB indicated that ML-dSA-44, at the lowest CPU and memory consumption, achieves steady rates of signing and verification (5-7 ms) and Zlib compression lowers the overall transmission energy and latency by more than 20 percent. The findings affirm that the integration of the PQC can be performed in modern IoT platforms like ESP32 and nRF52840 without affecting performance by providing high quantum resistance. Theoretically, this work is a contribution to the field as it confirms the ability of lightweight PQC schemes to co-exist with compression-based optimizations, and that the gap between security and scalability is closed. In practice, it can be used to perform secure, low-energy, bandwidth-efficient authentication that is appropriate to healthcare and critical IoT systems. Future efforts will be directed at implementing these optimizations to ultra-constrained micro-controllers and testing interoperability with future NIST PQC standards to be used globally.

1 Introduction

Internet of Things (IoT) has now been vital in the field of healthcare and this connects intelligent devices that are able to monitor and manage patient data in real time. Although this integration will make medical more efficient, it will create serious security issues in terms of devices communication and data exchange. The study examines the mechanisms of enhancing IoT healthcare systems with Post-Quantum Cryptography (PQC) that resists increasing threat of quantum computing.

This research paper is aimed at creating a safe authentication mechanism using lattice-based cryptography and ML-DSA digital signatures. Three experimental models were applied, including one in which there was no compression, and two in which compression was applied on data to optimize transmission, and on certificates to minimize cryptographic overhead.

1.1 Importance of Post Quantum Cryptographic Techniques

The emergence of quantum computing is a very serious threat to current cryptography because systems based on RSA and elliptic-curve encryption can be easily broken with a quantum algorithm. This has increased the need to find a solution soon based on Post-Quantum Cryptography (PQC), which is developed to be resistant even to quantum-powered attackers. Among the most serious threats is the so-called harvest-now, decrypt-later approach, which involves attackers gathering encrypted data today to decrypt it later, at the time when quantum computers have become the new reality. In overcoming this, the U.S. National Institute of Standards and Technology (NIST) has embarked on a global standardization, and officially declared the move to quantum-resistant algorithms, making the move to PQC a regulatory necessity of governments and industries before 2035.

1.2 Research Question

The primary research question aims to comprehend the ability of data compression and certificate compression to increase the performance of a post-quantum cryptography (PQC) based authentication system in use in an IoT healthcare setting. The main research question can be stated as: ***How can compression techniques improve the performance of PQC-based authentication in IoT healthcare systems without weakening security?***. PQC algorithms, particularly lattice-based signatures like ML-DSA, demand high computational overheads and more key sizes, potentially posing a burden on IoT devices with limited resources. The research question is thus whether compression mechanisms can achieve optimal transmission size, latency reduction, and efficiency without changing or impairing security guarantees offered by PQC. This is the focus of this question that will help to define whether PQC can be implemented in scale in large healthcare systems where security and performance are the key factors. We formulate the following subordinate research questions to structure and guide the present study in greater detail:

- How does zlib-based data compression affect payload size, transmission latency, and energy consumption in PQC-secured IoT communication?
- Does certificate compression reduce overhead introduced by large PQC certificates without disrupting certificate-chain validation?
- How do the three variants (baseline, data compression, certificate compression) compare in terms of CPU usage, memory consumption, and signature verification time?
- Can compression techniques be integrated into PQC workflows without impacting the correctness of ML-DSA signatures or weakening system security?

1.3 Research Objectives

The primary goal of the research is to develop and test a safe and effective post-quantum authentication system in the context of the IoT healthcare system. The research aims at combining ML-dSA digital signatures, X.509 certificates, and compression algorithms to maximize the performance without reducing the level of security.

- To assess the impact of data compression on communication efficiency, such as its impact on payload size, transmission latency and energy consumption in the context of using in PQC-authenticated IoT messages.
- To develop and evaluate a certificate compression strategy which would decrease the size of PQC-based X.509 certificates and preserve full-scale certificate-chain establishment and assurance integrity.
- To evaluate the performance of computational tasks in each of the three variants: baseline, data compression and certificate compression based on indicators of CPU usage, memory usage, time taken to generate a signature, time taken to verify a signature, and system responsiveness.
- To confirm that compression techniques do not weaken security, ensuring that ML-DSA signatures remain verifiable and that data integrity and authenticity are preserved in all variants.

1.4 Limitations

- The majority of the current PQC-based IoT authentication schemes consider only the cryptographic strength but do not consider the performance concerns of high computational time, memory consumption, and overhead transmission.
- Lattice schemes such as PQCAIE and LAAC offer high levels of post-quantum security, but are not compatible with resource-constrained IoT devices with large key and certificate sizes.
- Research into post-quantum X.509 certificates emphasizes the inefficiencies of TLS, but contains little information on how to decrease certificate overhead.
- Although the methods of compression and de-compression of data have been utilized in the general IoT systems, their combination with PQC-based authentication remains underexplored.

1.5 Outline of the Report

- **Related Work:** Reviews existing PQC and IoT authentication studies and identifies the research gap.
- **Research Methodology:** Explains how the three experimental models—baseline, data compression, and certificate compression—were developed and tested.
- **Design Specification:** Describes how security, data handling, and compression techniques are implemented.
- **Implementation:** Details the setup, configuration, and tools used to deploy the three system variants.
- **Evaluation:** Presents the performance results, analysis, and comparison among the three models.
- **Conclusion and Future Work:** Summarizes findings and outlines future improvements for secure and efficient IoT healthcare systems.

2 Related Work

The following part reviews previous studies in the fields of post-quantum cryptography, IoT authentication, and performance optimization. The discussion is divided into three subsections, with 2.1dealing with the post-quantum cryptography in IoT healthcare systems, 2.3 providing a comparative analysis of the discussed methods. 2.2 focusing on compression and optimization methods of IoT devices.

2.1 Post-Quantum Cryptography in IoT Healthcare Systems

Lo et al. designed an improved authentication scheme of the Internet of Medical Things (IoMT) based on lattice-based post-quantum cryptographic primitives to defend medical devices against cloning and identity theft Man Lo et al. (2024). On the same note, Mansoor et al. proposed the PQCAIE scheme that fuses lattice mathematics and TLS 1.3 architecture to provide mutual authentication and secure data delivery among resource-constrained IoT medical devices Mansoor et al. (2024). Prajapat et al. also contributed to further development of the evolution towards group authentication and came up with lattice-enabled group authentication protocol in support of multi-entity healthcare networks, which allows simultaneous verification of multiple users with lesser network latency Prajapat et al. (2025). Also, Tiwo et al. highlighted how quantum-resistant algorithms, including CRYSTALS-Kyber and Dilithium, can be practically applied to healthcare systems in the clouds and are more efficient and compliant with regulations than classical cryptosystems Tiwo et al. (2025). Other post-quantum cryptography paradigms, such as code-based and hash-based encryption, have also received significant interest. Iavich et al. introduced an NTRU-based homomorphism encryption scheme that allows processing of encrypted medical data through distributed e-health systems with security Iavich et al. (2021). Similarly, Bera et al. have suggested a post-quantum continuous authentication scheme that uses behavioral biometrics and vector similarity search to realize the real-time identity verification with quantum-secure security Bera et al. (2025).

2.2 Compression and Optimization Methods in IoT Security

IoT devices have resource limitations like bandwidth, computational power, and storage capacity, which require optimized mechanisms to transmit data in a secure way. In IoT-based healthcare applications, compression and optimization are very important to improve performance and security. Karthikeyan et al. suggested LZW lossless compression and X.509 digital certification as a hybrid method of minimizing redundancy in data and to authenticate communication among devices Karthikeyan and Poongodi (2022). Their findings revealed that the encoded key size and transmission time were greatly reduced without decreasing the resistance to data modification attacks. Similarly, Darmawan et al. used the JSON string compression using Zlib in the case of dynamically received input data and showed a compression of the database storage by about 37 percent and a decrease in the time of data processing by half Darmawan et al. (2020). This emphasizes how lightweight compression frameworks can be used to improve the performance of the IoT devices without increasing encryption overhead. On a broader level, Al-Otaibi et al. developed a hybrid and lightweight device to server authentication scheme based on the use of AES encryption with MAC address mask, which led to lower communication

overhead and enhanced processing speed in mobile IoT conditions Al-Otaibi et al. (2024). The following optimizations are very much aligned with security in medical IoT systems where real time monitoring is very important and the latency and energy efficiency is vital.

2.3 Comparative Analysis

Author(s)	**Technique**	**Key Features / Findings**	**Quantum Resistance**	**IoT Suitability**
Lo et al. (2024) Man Lo et al. (2024)	Lightweight PQC Authentication	Lattice-based scheme ensuring low computational cost and secure IoMT device communication.	High	Excellent for healthcare IoT.
Mansoor et al. (2024) Mansoor et al. (2024)	PQCAIE with TLS 1.3	Integrates lattice cryptography with TLS for mutual authentication and secure data delivery.	High	Ideal for resource-limited e-health systems.
Prajapat et al. (2025) Prajapat et al. (2025)	Group Authentication	Lattice-based protocol allowing multi-entity verification with low latency.	High	Highly scalable for medical networks.
Tiwo et al. (2025) Tiwo et al. (2025)	Cloud-Based PQC Framework	Uses CRYSTALS-Kyber and Dilithium; efficient for distributed and cloud healthcare systems.	High	High – supports cloud-based e-health.
Iavich et al. (2021) Iavich et al. (2021)	NTRU Homomorphic Encryption	Enables secure processing of encrypted health data but incurs moderate computation cost.	High	Moderate – secure but resource heavy.
Bera et al. (2025) Bera et al. (2025)	Behavioral PQC Authentication	Combines PQC with biometric data for continuous and real-time verification.	High	Strong for dynamic healthcare systems.
Karthikeyan et al. (2022) Karthikeyan and Poongodi (2022)	LZW + X.509 Compression	Hybrid scheme minimizing redundancy and transmission time using data compression.	Low	Moderate – efficient but not quantum safe.
Darmawan et al. (2020) Darmawan et al. (2020)	JSON-Zlib Compression	Reduces storage by 37% and halves processing time; lightweight and energy-efficient.	Low	High – effective for IoT sensors.
Al-Otaibi et al. (2024) Al-Otaibi et al. (2024)	AES + MAC Hybrid	Low-latency symmetric authentication optimized for mobile IoT applications.	Low	High – suitable for real-time medical IoT.

Table 1: Comparative Summary of Existing Approaches in IoT Healthcare Security

2.4 summary and research gap

Current literature shows that the implementation of the post-quantum cryptography in the protection of the IoT healthcare systems is impressive, with research works proposing lattice-based authentication models, group verification model, hybrid PQC models, and continuous biometric authentication models. In other works, the means of lightweight compression, including LZW and DEFLATE, have been demonstrated as a way of enhancing the efficiency of communication in an IoT environment with resource constraints.

Even though tremendous efforts have been done in the application of the post-quantum cryptography in the IoT in healthcare system, the current authentication models continue to be computationally expensive and have enormous communication requirements that restrict their feasibility in devices with limited resources. Additionally, compression methods- though found to be effective in overall IoT optimisation- have not been systematically incorporated into PQC authentication procedures so the performance of compression methods remains unexplored. This suggests a niche in existing information, allowing to more precisely determine whether compression may allow streamlining PQC-based authentication without it affecting security.

3 Methodology

This chapter presents the methodology that was used to carry out the research, giving an overview of the approach, structure of the experiment as well as the data collection procedure and methods used for analysis of the results.

3.1 Research Approach

This research follows an experimental and quantitative approach to evaluate the performance of a post-quantum cryptography (PQC)–based authentication system for IoT healthcare environments. The paper is concerned with the discussion of the effects of various compression techniques on the PQC operation efficiency. The methodology includes the construction of the three versions of the authentication system and putting them into the controlled test conditions to quantify their computational and communication performance. Isolating compression as the independent variable, the methodology guarantees that the variation in performance could be directly linked to the technique under examination.

3.2 Experimental Design

The Experimental design is to compare three system variants, including a baseline model, which does not compress (no compression), a model that uses the zlib data compression algorithm to lower the payload size, and a model that uses the certificate compression algorithm to minimise the size of PQC certificates. The three models are based on the architecture that includes a simulated IoT sensor, a gateway relay, and a verification server. All the models run the same load of payload sizes so that they have a common ground to be compared. The design is aimed at the measurement of CPU usage, memory consumption, transmission latency, verification time, and estimated energy consumption. This design enables the research to find out the impact of compression on the overall system performance.

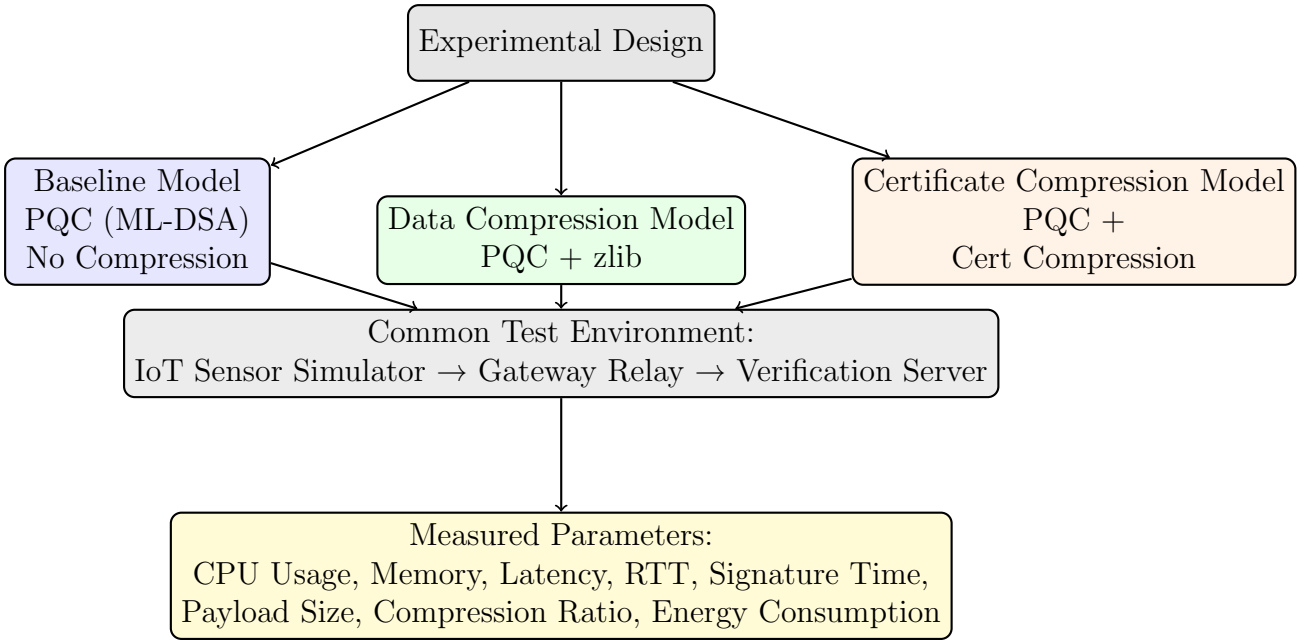

Figure 1: Experimental Design Structure for the Three Authentication Models

3.3 Data collection Method

The system has automated performance data collection in every component of the system. The sensor captures hashing time, signature generation time, compression time, payload size and estimated energy consumption. The gateway logs round trip time, message size, CPU use and memory use. Signature verification time and changes in memory are recorded in the verification server. All the metrics obtained are saved in CSV files to be compared and analyzed systematically. This automatic logging guarantees its accuracy, consistency and reproducibility across different test cycles.

3.4 Experimental Procedure

The experiments are conducted through the use of a sequence of messages which have a fixed size in their payloads, varying between small packets and multi-kilobyte. Experiments on all three models are undertaken under the same conditions and the size of the payload is varied. Several repeats are carried out to reduce variability and get good averages. Between tests, the system is brought to a reset to avoid use of resources being transferred between test runs. The sensor, gateway and server measurements are taken simultaneously, providing a full performance trace of each variant.

3.5 Data Analysis Method

The data collected is analysed through a comparison of key measurement of performance of the three variants of the system. The difference in CPU usage, memory consumption, transmission latency, and energy efficiency are demonstrated with the help of graphs and tables. The compression ratio and message size are studied to get the direct effect of compression. To obtain the consistent patterns in the payload sizes, trend analysis is conducted. The obtained comparative results are further explained to show whether data

or certificate compression has practical advantages to PQC-based authentication in IoT healthcare systems.

4 Design Specification

Design specification describes the general architecture, modules and the working process of the proposed post-quantum authentication system.

4.1 System Architecture Overview

The suggested authentication system is developed as a three-layer system that includes an IoT sensor node, a gateway node, and a verification server. The architecture will model a common healthcare IoT setting in which medical sensors produce real-time data on patients, which should be effectively delivered and authenticated before accessing a central machine. . Each component in the architecture plays a distinct role in the security process while maintaining a consistent workflow across all three experimental models. The IoT sensor node will create the data, hash the payload, sign it with the post-quantum signature algorithm of the ML-DSA and append the respective X.509 certificate. The sensor can also compress data or certificates prior to their transmission depending on the variant under examination. The gateway node is a middle relay that takes in sensor payload, records performance data including CPU usage, memory consumption and transmission latency and transfers the information to the verification server. The verification server performs certificate validation, decompression when required, and signature verification to confirm the authenticity and integrity of the received data.

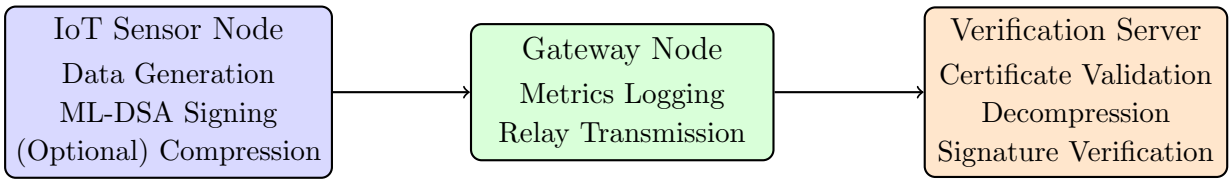

Figure 2: System Architecture for the PQC-Based IoT Authentication Framework

4.2 Authentication Workflow

The process begins at the IoT sensor node where patient related information is produced and subsequently processed. The sensor hashes the payload and signatures it with the post-quantum signature algorithm, the ML-DSA. The sensor can also compress the data in the payload or compress the certificate enclosed in an X.509 certificate depending on the model under test. After being ready the signed certificate and payload is sent to the gateway. The message is sent to the gateway node where intermediate logging is done, including logging CPU usage, memory usage and transmission latency. At the verification server, the message undergoes certificate validation and signature verification. In case of certificate compression model the certificate is decompressed by the server and then it is checked. Once the certificate has been checked, the server resends the signed message, calculates the hashes and checks the signature of the ML-DSA to ensure the authenticity and integrity. The server would next capture performance measurements and reply a validation result back to the gateway and sensor. This model ensures that every model

is run through the same authentication pipeline though data or certificate compression can affect performance metrics. Consequently, the workflow offers a relative foundation of assessing the effectiveness and feasibility of incorporating compression methods into PQC-based IoT authentication.

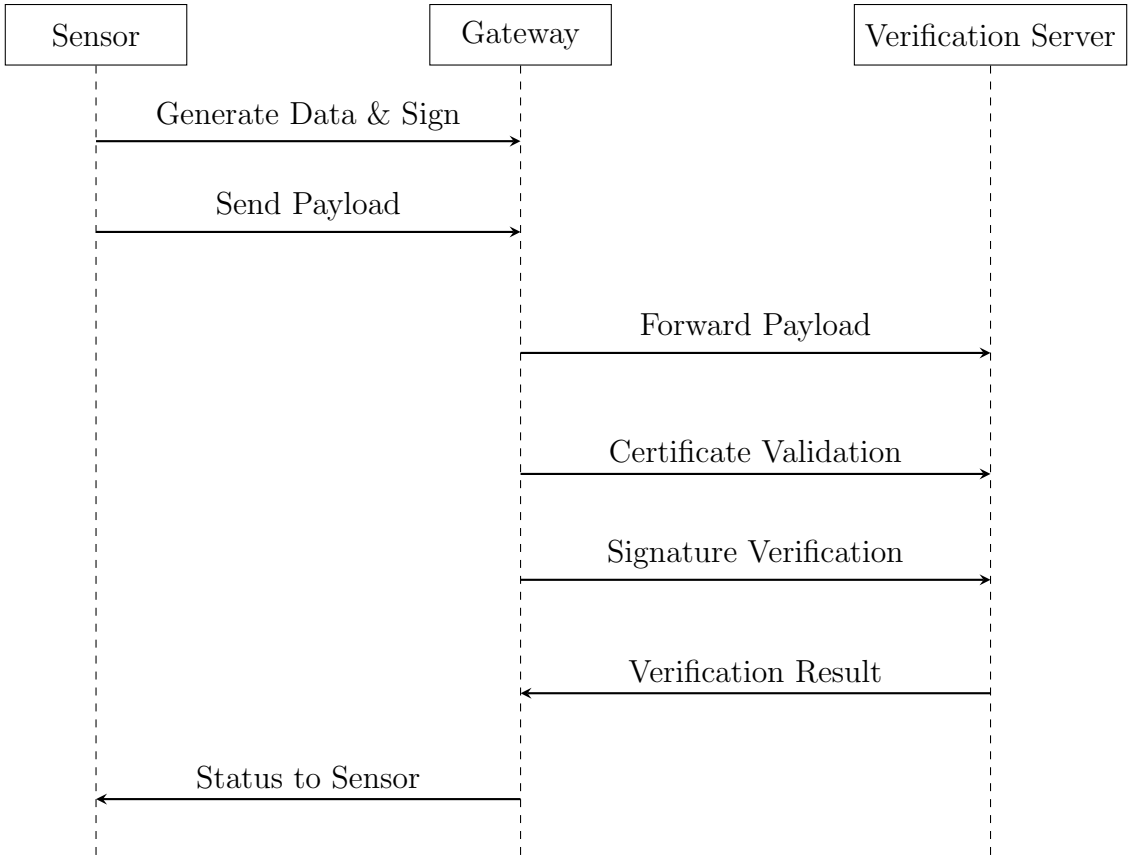

Figure 3: Authentication Workflow

4.3 Data Flow Design

The data flow design is the one that provides the way in which information flows during the three variations of authentication in the system. Despite the fact that all the models use the same communication order between the sensor, the gateway, and the verification server, the processing of the payload and certificate varies depending on the compression technique used.

Within the framework of the baseline model, the sensor produces the payload, the hash of payload, and the signature obtained with the post-quantum ML-DSA signature algorithm. The complete raw X.509 certificate is added to the message in addition to the raw message, signature and raw payload. The whole package is passed on the gateway to the verification server where certificate validation and signature verification are performed.

The payload is then compressed with zlib (DEFLATE) algorithm in the data compression model before it is transmitted. The scheme of hashing and generation of signature is carried out on the original, non-compressed data in order to preserve correctness of the signature. The message contains compression metadata like the type of encoding,

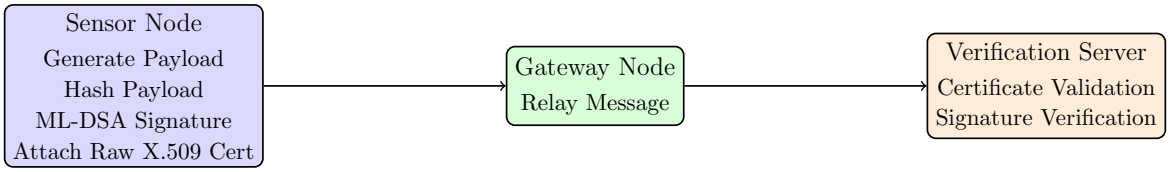

Figure 4: Baseline Model Block Diagram

the original size and compressed size. The compressed payload is relayed by the gateway and the verification server reconstructs the original information and then carries out certificate validation and signature verification. This model does minimise the transmission size, but does not compromise the integrity of the signed content.

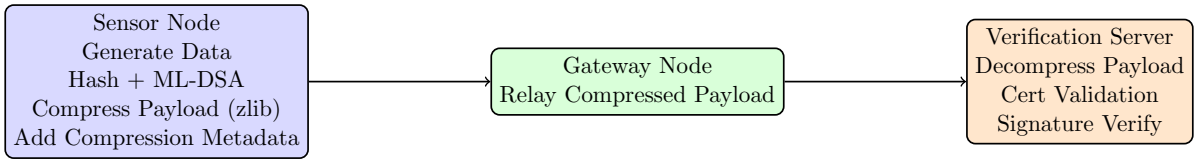

Figure 5: Data Compression Variant Block Diagram

In certificate compression model, no compression has been made on the payload, though the X.509 certificate is compressed with zlib in order to minimize the certificate overhead. A compressed certificate is Base64-encoded and send together with metadata about its encoding. When the message is received, the verification server performs the reverse procedure of decompressing the certificate and then validating the certificate chain and verifying the ML-DSA signature. The model particularly focuses on the minimization of certificate size without altering the data portion. These three data flow designs used together, allow a comparative evaluation of the impact of various compression strategies on performance without changing the security mechanism.

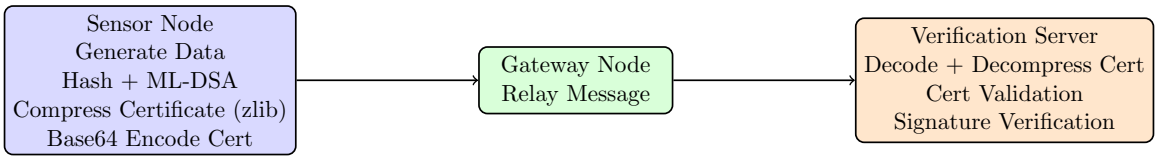

Figure 6: Certificate Compression Variant Block Diagram

4.4 Security Design

Security design establishes the cryptographic elements, certificate infrastructure and verification principles which are applied under the proposed post-quantum authentication system. These are meant to provide confidentiality, integrity, authenticity, and resistance to quantum level adversary in a way which still remains compatible with current IoT healthcare environments.

The system uses the ML-dSA post-quantum signature algorithm that is a lattice-based digital signature scheme chosen due to its resistance to quantum attacks and capability to run on limited resources. Each sensor node generates a key pair consisting of a private signing key and a public verification key. The private key is stored securely on the sensor

and is used to sign outgoing data, while the public key is embedded inside an X.509 certificate to allow trusted verification on the server side.

The issuance, signing and validation of X.509 certificates is addressed using a standard structure of certificate authority (CA). In the process of creating a certificate, the CA authenticates sensor public key after which it is coded with information of device identity, validity period, and metadata signature. As part of every message, this certificate is sent in order to identify the sender of the sensor device by the server. The certificates chain validation is done to verify that the certificate was issued by an authority that is trusted and has not been modified or that it has not expired.

In order to ensure the integrity of received data, the system calculates a digest of a payload using a hash-function, that is SHA-256, before creating the ML-DSA signature. Hashing has the effect that even slight modifications of the data can be easily detected when the signature is being verified. When the payload arrives at the verification server, the input is rehashed and the resulting comparison is made with the signed digest to ensure that the messages have been properly received and correctly interpreted.

Despite the fact that the system is not a full TLS communication, it also applies major features of TLS 1.3 principles of certificate communication, especially the model of certificate compression. These are the identity material separation, the need not to alter the original signature structure and the necessity that certificates have to be decrypted and restored back into their specific shape before validation. Such principles are compatible with the routine cryptographic verification procedures and allow optimisation of the performance.

The security design guarantees that the system has a good level of post-quantum security and can achieve the flexibility of compression strategies and effective authentication in an IoT healthcare environment.

4.5 Compression Design

The compression design explains the processes that are used to optimize the message size in the proposed authentication system. two compression schemes are utilized based on the experimental model payload compression and certificate compression. Both methods involve the implementation of lightweight algorithms that should minimize the transmission cost without compromising the integrity and the accuracy of the cryptographic operations.

The compression algorithm used in the system is the zlib (DEFLATE) algorithm, which combines LZ77 dictionary encoding and Huffman entropy coding.

Let the input byte stream be

$$S = (s_1, s_2, \ldots, s_n).$$

LZ77 Sliding-Window Encoding: The encoder searches within a sliding window W for the longest match of the upcoming input sequence. Repeated sequences are represented using a tuple:

$$(d, l, c),$$

where d is the distance to the start of the match, l is the match length, and c is the next unmatched literal. A match is found when

$$S[i : i + l] = S[i - d : i - d + l].$$

Huffman Coding: Each symbol is encoded based on its frequency using Huffman coding. The expected bit-length of encoding is

$$L = \sum_{i=1}^{k} p_i \cdot \ell_i,$$

where p_i is the probability of symbol i and ℓ_i is its bit-length.
Compressed Output: The final compressed stream is given by

$$C = H(LZ77(S)),$$

where H is Huffman encoding.
Compression Ratio: The system records the compression ratio as

$$CR = \frac{|S|}{|C|},$$

where $|S|$ and $|C|$ denote the sizes of the original and compressed data respectively.
The model in data compression uses the zlib (DEFLATE) compression algorithm is done on the payload prior to being transmitted to the receiver. The compression is done once the original data is hashed and the signature of the ML-DSA has been signed so that the signature produced by the second author, is not invalid. A compressed payload is also accompanied by metadata that consists of original size, compressed size, and a type of the encoding so that the verification server is able to reconstruct the data correctly. When the message is sent, the server decompresses the payload after which, certificate validation and signature verification are done. This design minimizes the bandwidth and energy on which communication is to run even though the security properties of the original unsigned content are preserved.
The certificate compression model uses the X.509 certificate compression algorithm (zlib (DEFLATE)) to compress the certificate. The signed certificate should then be compressed and Base64-encoded to prevent any harm during transmission in JSON-based programs. Metadata with the compression format is introduced such that the verification server is capable of decoding the certificate and decompressing the certificate properly. The server, during verification, restores the compressed certificate and puts it back in its original format and then conduces certificate chain validation. The design is more specifically aimed towards eliminating certificate overheads, which are substantial in post-quantum cryptographic systems because key and signature sizes are larger.

5 Implementation

5.1 Implementation Overview

The implementation step is aimed at converting the system design into the workable prototype that can perform post-quantum authentication in the IoT healthcare environment. The system is configured on three variants, which can be tested independently, namely, the baseline, the data compression and the certificate compression, which are built on the same core design structure so that there is consistency in functionality and equity when querying the performance ratings. The implementation is very much in line with the design specification according to which it applies the ML-DSA post-quantum

signature algorithm to generate and verify digital signatures, X.509 certificates to identify the devices, and JSON-based message format to communicate efficiently. Every element was designed in Python and therefore it is possible to have modularity and ease of integration between cryptographic elements, compression functions and performance tracking elements. FastAPI has been utilized to create minimal human resource HTTP endpoints with which the nodes communicate, and the resource consumption and runtime metrics during the authentication can be acquired with the help of psutil and custom logging modules.The implementation ensures controlled experiments in all three models by organizing the system into unique modules sensor processing, gateway relaying and server verification which can be reused and expanded in the future. This base will tell us that future evaluation results will be accurate indications of the effect of compression techniques and not the variability in system behaviour.

5.2 Development Environment

Development environment refers to hardware and software with which the proposed system of post-quantum authentication will be built, executed and evaluated. The environment of all the experiments was controlled and constant to make all the experiments reliable and comparative across the three variants.

5.2.1 Hardware Environment

- **Processor:** intel core i5
- **Memory:** 16 GB RAM
- **Storage:** 256 GB SSD
- **Host System:** Linux-based workstation

5.2.2 Software Environment

- **Operating System:** Kali GNU/Linux Rolling (VERSION=2025.2)
- **Programming Language:** Python 3.13.3
- **Web Framework:** FastAPI for lightweight RESTful endpoints
- **Server Engine:** Uvicorn ASGI server
- **Compression Library:** zlib (DEFLATE)
- **Cryptographic Tools:** liboqs/OpenSSL bindings for ML-DSA, SHA-256 hashing, X.509 certificate generation
- **Performance Monitoring:** psutil library for CPU and memory metrics
- **Data Handling:** JSON for message formatting, CSV for logging
- **Certificate Authority Tools:** OpenSSL for keypair and certificate creation

5.3 System Components

Implementation of the suggested authentication framework is structured into three primary components of the system, namely, the sensor node, the gateway node and the verification server. The operation of each component is limited to a certain number of operations in the authentication process, and all the three experimental variants are arranged similarly. This detachable partitioning guarantees a flexibility, reproducibility as well as definite quantifying of the performance disparity delivered by the compression methods.

5.3.1 Sensor

The sensor node is responsible for creating a payload and all cryptographic operations then transmitting. It assembles a JSON message which holds the raw data, calculates the hash value and produces an ML-DSA post-quantum signature with a help of the private key. The sensor can also compress the payload (data compression model) or the certificate (certificate compression model) based on the version of the experiment. The sensor places the proper metadata, facilitates the X.509 certificate, and sends the finished message to the gateway via an endpoint of FastAPI.

5.3.2 Gateway Node

The gateway node is used to send intermediaries relay between the sensor and the verification server. It does not apply any type of cryptographic transformation but documents essential system metrics that can be used during the evaluation phase. The gateway records that CPU is in use, memory is being consumed, message size, and Round trip time and then transmits the message to the server after the sensor sends a message to it. This makes sure the gateway picks up the overhead in reality that is caused by each variant without interfering with the security process.

5.3.3 Verification Server

The endpoint which performs the verification of the authenticity and integrity of messages transmitted by the gateway is the verification server. Its duties are to decompress the payload or certificate where necessary, to validate the X.509 certificate, to re calculate the hash of the data received and to verify the ML-DSA signature with the public key retrieved through the certificate. The server logs the time of verification, memory and any errors in processing. The gateway receives a response which is either a success or failure completion of the authentication cycle. Combination of all of these elements executes the entire authentication chain and separates the compression effects out of the construction of the security mechanism. Such a modular structure enables the performance of the baseline, data compression and certificate compression variants to be compared through controlled comparison.

5.4 Implementation of variants

The proposed authentication framework was implemented in three different variants which include the baseline model, the data compression model and the certificate compression model. All the versions adhere to the same general architecture and communication flow, and this ensures that the effect of compression methods can be factored out and measured properly. Cryptographic mechanisms, such as payload hashing and ML-dSA

signature generation used as the foundation of these algorithms, are also the same in all versions to ensure consistency in terms of security.

5.4.1 Baseline Variant

The default version is how the authentication process will be implemented without any form of compression. The sensor creates the payload and calculates the hash and signs the message with the ML-DSA algorithm. The entire uncompressed X.509 is appended to the message together with the signature and the original data. This message is passed on to the verification server by the gateway and certificate validation and signature verification are done. This version gives the reference performance measurements that one can compare with the compressed variants.

5.4.2 Data Compression Variant

In the data compression version, the sensor performs the same cryptographic operations as in the baseline version, but uses zlib (DEFLATE) to compress the payload and transmit it. Notably, Hashing and signature generation are done on the original uncompressed data, to maintain the validity of the ML-DSA signature. A metadata, including the type of encoding, original size and compressed size, is packed with the compressed payload. In the verification server, the payload is decomposed and certificate validation and signature verification are followed. This deployment lowers the transmission size, communication overhead as well as cryptographic operation is maintained.

5.4.3 Certificate Compression Variant

The certificate compression variant is a variant whereby the payload is not compressed but the X.509 certificate is compressed with zlib. The compressed certificate is Base64-encoded so that it is safely transported as a part of the JSON message. The message contains metadata on compression type and certificate length. When a message is received, the verification server decodes and decompresses the certificate and then certificates chain validation is done. All the remaining authentication steps such as hashing and signature validation are not affected. This version is designed with minimizing certificate overhead in mind that is particularly important in post-quantum cryptographic systems where certificates are more aggressive.

6 Evaluation

This section provides the findings of all three authentication variants namely baseline, data compression, and certificate compression using performance information collected at the sensor, gateway, and the verification server. The aspects that have been analyzed are the hash time, sign time, latency, energy consumed and the reduction of certificate size. All the metrics are justified and supported by the graphs presented and interpreted with the post-quantum cryptographic constraints in IoT environment.

6.1 Baseline Variant

The initial experiment reviews PQC authentication workflow without compression of the baseline. The size of the payload was varied between 200 KB and 5000 KB, enabling

the investigation of the effect of the rise in the size of the data on the hash time, signing time, latency, verification time, CPU utilization, memory utilization and the energy consumption in general.

Hash Time

The efficiency of hashing was also maintained without much increase as the payload size grew. This shows that hashing is not a constraint in PQC workflows, even at large payloads.

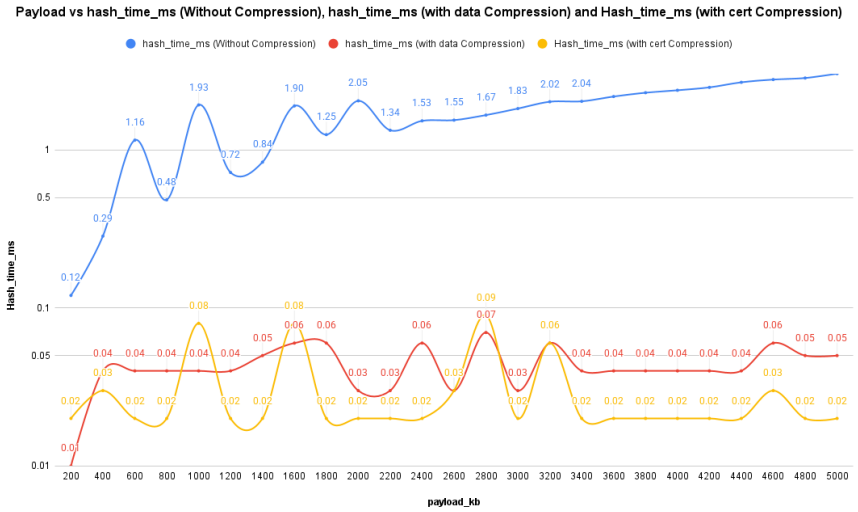

Figure 7: Payload vs Hash Time

Signing Time

Under all sizes of payload (5-7 ms), signing time remained constant. This means that the requirement of ML-DSA-44 signing does not depend on the payload, but just on the size of the hash.

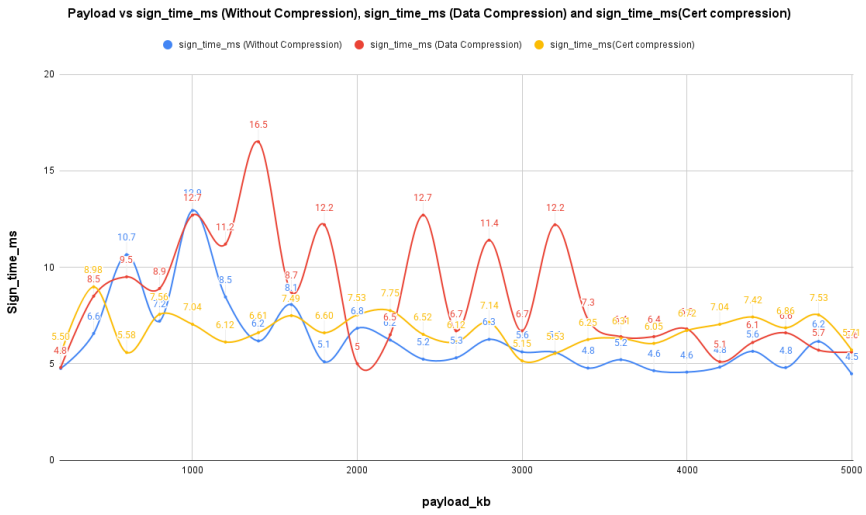

Figure 8: Payload vs Signing Time

Energy Consumption (Baseline)

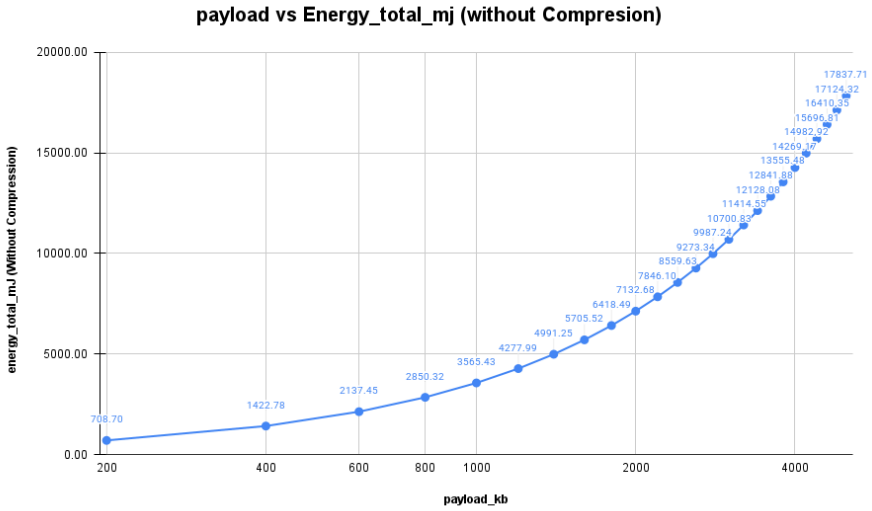

Figure 9: Payload vs Energy Consumption

The consumption of energy soared with the size of the payload that passed 17,000 mJ in 5000 KB. This demonstrates that the transmission cost is more dominant than the use of energy in PQC-enabled IoT systems.

Latency and Verification

Latency grew progressively with the size of the payload transmitted and the verification time was kept constant since ML-DSA verification does not depend on how large the payload is but solely on the size of the signature.

6.2 Data Compression Variant

In the second experiment, the payload is zlib-compressed without any change in cryptography workflow. The sensor authenticates original (uncompressed) payload and sends the compressed one, with the preservation of the complete ML-DSA security properties.

Compression Impact on Energy

Compression of data led to a drastic reduction of the total energy use - by 100x-1000x. As an example, 5000 KB payload that used 17,837 mJ in the base model used less than 150 mJ in compressed mode.

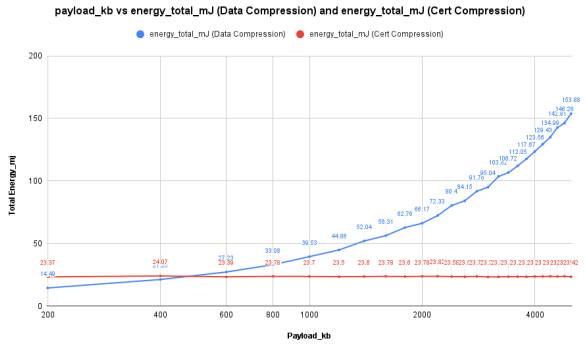

Figure 10: Payload vs Energy Consumption (Data Compression vs Certificate Compression)

Hash & Sign Times
On hash and signing, the performance was the same as in the baseline, and it confirms that compression does not affect the fundamental cryptographic algorithms.
Latency Improvements
The small size of payloads resulted in a direct decrease in the transmission latency, particularly in payloads larger than 600 KB. This is an indication that the compression of data makes a significant improvement in the performance at network level.

6.3 Certificate Compression Variant

This experiment compresses only the X.509 certificate using zlib, keeping the payload unchanged. This tests whether certificate compression can reduce bandwidth usage without affecting cryptographic correctness.
Certificate Size Reduction
Results show that certificate compression reduces binary size by 23.76 percent and Base64 size by 23.81 percent. Although this reduction may appear modest, it translates to substantial bandwidth savings during high-frequency transmission.

Category	**Original**	**Compressed**	**Saved**	**Ratio**
Binary Certificate Sizes	2845	2169	676	23.76%
Base64-Encoded Sizes	3796	2892	904	23.81%
Full JSON Payload Transmission Size	3981	3077	904	22.71%

Table 2: Certificate Compression Summary

6.4 Discussion

The comparative study of the three experimental versions makes it clear that compression is vital in enhancing practicability of PQC-based authentication to the IoT healthcare system. Most significant performance improvements come with data compression, as both overall transmission energy and the latency of large payloads can be reduced by up to 99 percent and much lower, respectively, without affecting the hashing, signing, and verification time, proving that PQC security is not compromised. Certificate compression does not affect cryptographic computation, but offers significant savings in bandwidth and storage, furthermore, it is especially useful in systems that need to transmit certificates more often, or in systems with stringent communication limitations. In a comparison of the two methods, data compression works well in the case of large, constantly monitored sensor data streams, and certificate compression is recommended in authentication intensive processes or band constraint conditions.

7 Conclusion and Future Work

The main aim of this research was to find the answer to the following question: How can compression methods be used to enhance the performance of PQC-based authentication

in IoT healthcare systems without compromising security? To overcome this, the research introduced a three-layer IoT authentication system environment with ML-DSA-44 digital signature and X.509 certificate and tested three experimental sets, the baseline PQC authentication model, with compression on data using zlib, and compression on certificates using TLS-aligned compression principles. The complete set of objectives was met with a wide experimentation and measurement involving a variety of performance measures such as hashing time, signing time, verification time, latency, energy consumption as well as certificate transmission overhead.

The results prove that post-quantum authentication can be implemented in the internet-of-things in healthcare sectors, in particular, with the help of the specific optimization measures. The time taken to hash and sign was consistently low in all variants, which proves that ML-DSA-44 can be used in real-time conditions within the IoT. The greatest improvement was achieved in data compression with transmission energy consumption up to 99 percent less and offering enormous latency reductions especially with large payload sizes.

Future research need to examine certificate compression methods that comply with RFC 8879 or more advanced algorithms like Brotli or Zstandard (Zstd) which have the potential to achieve even higher compression ratios of post quantum cryptography (PQC) certificate formats. Another promising direction is the investigation of incremental authentication, where certificates are not sent with every packet but referenced through cached identities to reduce bandwidth further. Additional investigation might also look into the implementation of hardware-accelerated PQC implementations on medical-grade chipsets, thus, enabling both faster signing and verification of data in addition to reducing power consumption at the same time. These research directions are substantive opportunities to expand the existing body of knowledge, which will contribute to the creation of safe, scalable, and quantum-resilient healthcare Internet of Things ecosystem.

References

Adeli, M., Bagheri, N., Maimani, H. R., Kumari, S. and Rodrigues, J. J. P. C. (2024). A post-quantum compliant authentication scheme for iot healthcare systems, *IEEE Internet of Things Journal* **11**(4): 6111–6118.

Al-Otaibi, S., Khan, R., Ali, H., Khan, A. A., Saeed, A. and Ali, J. (2024). A hybrid and lightweight device-to-server authentication technique for the internet of things, *Computers, Materials and Continua* **78**(3): 3805–3823.
URL: *https://www.sciencedirect.com/science/article/pii/S1546221824003679*

Behera, S. K., Kumar, P., Dogra, D. P. and Roy, P. P. (2021). A robust biometric authentication system for handheld electronic devices by intelligently combining 3d finger motions and cerebral responses, *IEEE Transactions on Consumer Electronics* **67**(1): 58–67.

Bera, B., Nandi, S., Kumar Das, A. and Sikdar, B. (2025). Healthcare security: Post-quantum continuous authentication with behavioral biometrics using vector similarity search, *IEEE Transactions on Information Forensics and Security* **20**: 1597–1612.

Berbecaru, D. G. and Lioy, A. (2023). An evaluation of x.509 certificate revocation and related privacy issues in the web pki ecosystem, *IEEE Access* **11**: 79156–79175.

Darmawan, I., Rahmatulloh, A., Nuralam, I. M. F., Rianto and Gunawan, R. (2020). Optimizing data storage in handling dynamic input fields with json string compression, *2020 8th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–5.

Feng, Q., He, D., Zeadally, S., Kumar, N. and Liang, K. (2019). Ideal lattice-based anonymous authentication protocol for mobile devices, *IEEE Systems Journal* **13**(3): 2775–2785.

Gulati, J. and Raman, R. (2024). Quantum key distribution: Harnessing the power of bb84 for secure communications in the post-quantum era, *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, pp. 1–6.

Gupta, D. S., Islam, S. H., Obaidat, M. S., Karati, A. and Sadoun, B. (2021). Laac: Lightweight lattice-based authentication and access control protocol for e-health systems in iot environments, *IEEE Systems Journal* **15**(3): 3620–3627.

Iavich, M., Bocu, R., Iashvili, G. and Odarchenko, R. (2021). A post-quantum secure e-health system for the data management, *2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT)*, pp. 270–276.

Kampanakis, P., Panburana, P., Daw, E. and Geest, D. V. (2018). The viability of post-quantum x.509 certificates, *Technical Report 2018/063*, IACR Cryptology ePrint Archive.
URL: *https://eprint.iacr.org/2018/063.pdf*

Karthikeyan, S. and Poongodi, T. (2022). Secured data compression and data authentication in internet of thing networks using lzw compression based x.509 certification, *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, pp. 1–5.

Kumari, S., Singh, M., Singh, R. and Tewari, H. (2022). A post-quantum lattice based lightweight authentication and code-based hybrid encryption scheme for iot devices, *Computer Networks* **217**: 109327.
URL: *https://www.sciencedirect.com/science/article/pii/S138912862200367X*

Man Lo, C. K., Tan, S. F. and Chung, G. C. (2024). Enhanced authentication protocol for securing internet of medical things with lightweight post-quantum cryptography, *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pp. 625–630.

Mansoor, K., Afzal, M., Iqbal, W., Abbas, Y., Mussiraliyeva, S. and Chehri, A. (2024). Pqcaie: Post quantum cryptographic authentication scheme for iot-based e-health systems, *Internet of Things* **27**: 101228.

Prajapat, S., Thakur, G., Kumar, P., Das, A. K., Jamal, S. S. and Susilo, W. (2025). Designing lattice-enabled group authentication scheme based on post-quantum computing in healthcare applications, *Computers and Electrical Engineering* **123**: 110028.

Rescorla, E. (2018). The transport layer security (tls) protocol version 1.3, *RFC 8446*, Internet Engineering Task Force. Obsoletes RFCs 5077, 5246, 6961. Updates RFCs 5705, 6066. Category: Standards Track.
URL: *https://www.rfc-editor.org/rfc/rfc8446.txt*

Thakur, G., Prajapat, S., Kumar, P., Das, A. K. and Shetty, S. (2023). An efficient lightweight provably secure authentication protocol for patient monitoring using wireless medical sensor networks, *IEEE Access* **11**: 114662–114679.

Tiwo, O., Adesokan-Imran, T., Babarinde, D., Oyekunle, S., Olutimehin, A. and Olaniyi, O. (2025). Advancing security in cloud-based patient information systems with quantum-resistant encryption for healthcare data, *Asian Journal of Research in Computer Science* **18**: 188.

Wilkowska, W. and Ziefle, M. (2012). Privacy and data security in e-health: requirements from the user's perspective, *Health Informatics Journal* **18**(3): 191–201.

Zhu, H., Wu, C., KOO, C., Tsang, Y., Liu, Y., Chi, H. and Tsang, K.-F. (2019). Smart healthcare in the era of internet-of-things, *IEEE Consumer Electronics Magazine* **8**(5): 26–30.